

10/531217
JC12 Rec'd PCT/PTC 1 4 APR 2005
PATENT APPLICATION
ATTORNEY DOCKET NO.: 11345/120001

APPLICATION
FOR
UNITED STATES LETTERS PATENT

**TITLE: DISPLAY SCREEN CAPABLE OF BEING SUBJECTED
 TO A SCROLL PROCEDURE**

APPLICANT: Jean-Stéphane VILLERS

22511
PATENT TRADEMARK OFFICE

"EXPRESS MAIL" Label No.: EV562271571US

Date of Deposit: April 14, 2005

**DISPLAY SCREEN CAPABLE OF BEING SUBJECTED TO A SCROLL
PROCEDURE****DESCRIPTION**

5

FIELD OF THE ART

The present invention relates to a method of displaying
a document, for example an HTML (Hypertext mark-up
10 language) page, on a display screen capable of being
subjected to a scroll procedure, for example on a
digital television screen.

STATE OF THE PRIOR ART

15

The field of the invention is in particular that of the
displaying of HTML pages, as described in patent
application EP 1 164 499.

20 A version of the HTML standard ("HTML 4.01
Specification W3C Recommendation 24 December 1999") may
be found at the following Internet address
<http://www.w3.org/TR/1999/REC-htmT401-1999> 1224.

25 An HTML page is a document which can be interpreted by
a reading program to produce a visual, and possibly
audio, output on a computer monitor or on a television
screen. An HTML page presented on a screen ordinarily
consists of a main document and of secondary documents
30 that may contain graphical or sound elements, or source
code.

In order to take account of this HTML page, these
elements are loaded, stored in memory and processed by
35 an HTML engine. The representation of this page is
stored in a graphics memory for display.

- 2 -

An HTML engine is an interactive application making it possible to display hypertext and multimedia documents that can be accessed through the Internet network. It also makes it possible to send and receive electronic
5 mail and to consult so-called "news" servers.

In the field of digital television an interactive application such as this is driven by the user by means of a remote control or with the aid of an infrared
10 keypad. The application may, in fact, integrate a substitution interface allowing the user to perform all the entries of text or the like with the remote control alone. The displaying of the documents is carried out on the screen.

15 The retrieval of the documents on the Internet network is performed by way of queries on particular servers. In the field of digital television this operation may be asymmetric. Specifically, the query and the document
20 do not necessarily travel through the same medium: the query may be made through the telephone network via a modem, and the document may return either through the antenna, that is to say through satellite transmission, or through the modem. The user can then choose the type
25 of return.

The content of an HTML page may exceed the space available on the monitor or on the television screen. In this case the monitor or the television screen
30 displays only a part of the HTML page. The user can then scroll the display to get a glimpse of the remaining parts of the HTML page.

After scrolling, the new visible part of the HTML
35 document must be displayed. Each graphical element intersecting with the new visible part must be drawn or redrawn.

The scrolling of a document is commonly carried out in several ways. The simplest fashion is display without prior preparation in a buffer graphics memory, that is to say by drawing directly in the window displayed on the screen. Should the document be scrolled, the content of the window always remaining visible is displaced by a mass copy to its new position in the window. The remaining band, newly appeared, is displayed by individually redrawing each graphical object. This method has the drawback, when the graphical system is slow, of seeing the drawing of each of the graphical objects be composed progressively.

15 To remedy this problem, a solution used is to prepare the part of the document to be displayed in a buffer graphics memory such as for example a pixmap and to display in one go the content of this pixmap in the visible window. Here two techniques are used. The first

20 consists in having this buffer memory cover only the visible zone of the document. In this case, when a scroll is requested the new visible portion of the document is prepared in the buffer memory and when the latter is ready, it is displayed (technique called

25 "double buffering"). The second consists in covering the entire document with a buffer graphics memory, preparing the entire document and displaying it as and when required.

30 The major problems of these two techniques are that, in the first, the display is prepared upon an explicit scroll request. Consequently, a certain time that is a nuisance for the user may occur between the request and the actual scrolling, the scrolling not being fluid. In

35 the second, though the problem of waiting time does not exist, it may require a large quantity of graphics memory since the latter is related to the size of

- 4 -

documents. On the Internet network, the size of the documents is not limited and the quantity of graphics memory required in order to use this second technique may be considerable.

5

Such a solution is not, either, suitable for the display of a decoder since the latter requires a sizable memory to store the pixmap data which are calculated before display. Now, the memory capacity of a television decoder is limited. It is for example of the order of 2 megabytes of graphics memory. It is therefore appreciably more limited than that of a computer which may be 32 megabytes, or 64 megabytes, etc.

15

An objective of the invention is to solve such a problem, that is to say to offer fluid scrolling of a document in a context of limited memory, this document possibly for example being an HTML page.

20

DESCRIPTION OF THE INVENTION

The present invention relates to a method of displaying a document on a display screen capable of being subjected to a scroll procedure, characterized in that it comprises the following steps:

25

30

35

- a step of allocating the document a quantity of graphics memory so as to create a buffer memory of the visible part of the document and of the zones closest to this visible part and referred to as "anticipation bands",
- a step of calculating and of chopping of this memory into pixmaps as a function of the size of the document, of the visible part, and of those of the anticipation bands,
- a step of relative positioning of these pixmaps with respect to the complete document and its visible

- 5 -

part,

- a step, that can be carried out as a background task, of filling the content of the pixmaps with a priority system dependent on the proximity of the pixmap with respect to the visible zone,
- when the document is subjected to a display procedure or to a scrolling, a step of copying the content of the pixmaps into the display window with previously if necessary a step forcing the updating of the pixmaps involved in the display if the previous step has not terminated same,
- and return to the step of relative positioning of the pixmaps with respect to the document as a function of the new position of the visible part.

15

Advantageously in the case of horizontal and vertical scrolling, the anticipation bands comprise a minimum of one column of pixmaps on the right and on the left of the visible window as well as a row of pixmaps at the bottom and at the top, except in the case where the visible window approaches the edge of the document. The pixmaps are chopped into rectangles which are drawn successively with each call of a background task.

20

The filling of a pixmap if it is of large size may be lengthy and disable the system for the time of the operation. The filling of the pixmaps by rectangles of small sizes makes it possible to dilute the filling operation among the other processing operations of the system, and to be able to respond rapidly to the user if necessary.

30

The background task also has the function of constructing the anticipation zone.

35

With each call of this background task, the process is as follows:

- 6 -

- possible reorganization of the pixmaps if a scroll has been performed,
- if no repositioning of the pixmaps has occurred, drawing of the first rectangle of a pixmap determined as a function of a criterion of distance away from the visible zone of the document.

Advantageously the method of the invention uses a synchronization mechanism allowing the possible forcing of the data to be displayed into the pixmaps.

An immediate drawing is carried out in two cases:

- when an "expose" event compels the drawing of a part of the display window though this part has not yet been drawn in the anticipation bands,
- when an element of the document is modified graphically in the display window.

The method of the invention may in particular be used for the display of an HTML document, and/or for a digital television decoder.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 illustrates the displaying of the anticipation zone according to the method of the invention.

Figure 2 illustrates an exemplary displacement of the anticipation zone.

Figure 3 illustrates the synchronization mechanism, when a part of the pixmaps is not filled and when a display is requested.

Figure 4 illustrates the makeup of a tree of subwindows in an exemplary implementation of the method of the invention.

- 7 -

Figure 5 illustrates the display of the tree represented in figure 4.

- 5 Figures 6 and 7 each illustrate an HTML document after page setup according to the method of the invention; the complete width of the document being coverable with the pixmaps for figure 7, while for figure 6 this is not possible.

10

DETAILED DESCRIPTION OF PARTICULAR EMBODIMENTS

The method, according to the invention, of displaying a document, for example an HTML page, on a display screen
15 capable of being subjected to a scroll procedure consists in chopping this document into zones that may be covered by pixmaps whose size is dependent on the size of the display zone, and in providing a zone anticipating part of the document actually covered by
20 pixmaps.

This anticipation zone forms a matrix of pixmaps, covering the document, the visible part and the closest bands around this visible part, in such a way as to
25 prepare the content of the visible part and the zones that will soon be visible should the document be scrolled.

The purpose of the anticipation zone is to improve the
30 visual rendition of the operations of redrawing a part of the document. Specifically, in most cases, a copy of the anticipation zone in the display window turns out to be sufficient.

35 Figure 1 represents such an anticipation zone 10, here formed of 18 rows and 12 columns of pixmaps 13, the visible window 11 where the document 12 is displayed.

Note the hierarchical inclusion of the three zones thus formed, the document 12 including the anticipation zone 10, the latter containing the visible window 11 of the document. The anticipation zone may indeed cover the document widthwise or heightwise if the space available is sufficient, as shown by the illustration of figure 7.

Figure 2 shows the evolution of the anticipation zone 10 following an operation of scrolling the display of the document 12 downwards. This scrolling operation entails a displacement of the first row at the bottom of the anticipation zone. This row must then be redrawn.

The anticipation zone comprises a minimum of one column of pixmap zones on the right and on the left of the visible window as well as a row of pixmap zones at the bottom and at the top, except in the case where the visible window 11 approaches the edge of the document 12. If this rule is not complied with following a scroll, the anticipation zone displaces in such a way as to reestablish the latter.

So as not to risk disabling the application for several seconds, the drawing of the anticipation zone, following its creation or a scroll, is such that a background task performs this drawing in several steps: the pixmap zones are formed as rectangles that are drawn successively on each call of the background task.

This background task also has the function of constructing the anticipation zone.

With each call of this background task, the process is as follows:

- 9 -

- displacement if necessary of the anticipation zone by permutation of pixmap rows or columns,
- if the anticipation zone is correctly positioned, drawing of the first rectangle of a pixmap of the
5 anticipation zone.

If the pixmap to be displayed is not ready at the moment of display, a synchronization mechanism makes it possible to force the refreshing of the data to be
10 displayed in the pixmaps.

An immediate drawing is carried out in two cases:

- when an "expose" event compels the drawing of a part of the display window though this part has not yet
15 been drawn in the anticipation zone,
- or when an element of the document is modified graphically in the display window.

These two cases correspond respectively to an influx of new data relating to an image and to the processing of
20 the focus which modifies the drawing of an element of the image.

In both these cases all the waiting rectangles having a common intersection with the zone to be drawn are
25 drawn. Figure 3 represents the elements which then make up the anticipation zone:

- the rectangles already drawn 20,
- the undrawn rectangles 21,
- 30 - the display window 22,
- the rectangle 23 to be drawn following an "expose" event,
- the undrawn rectangles 24 which interact with the rectangle 23, the drawing of their content being
35 forced by the synchronization.

Implementation of the method of the invention in an

- 10 -

HTML application

We shall now consider an exemplary implementation of the method of the invention, in an HTML application in
5 a decoder.

The first operation consists in dividing up the usable graphics memory into a tree of subwindows (or HTML "frames").
10

The quantity of graphics memory available to create the pixmaps is limited at the decoder level. To guarantee that the other applications that are functioning at the same time as the HTML engine function properly, only a
15 part of this memory is used by the HTML engine to create the anticipation zones. The quantity of memory available is defined when the HTML engine is launched. In the case where several drop-down documents are visible simultaneously on the screen (case of HTML
20 subwindows or "frames"), each document must be able to profit from the fluid scrolling mechanism that the method of the invention constitutes. The quantity of memory available must therefore be divided up between all the documents that are visible simultaneously.

25 The method of the invention relates only to drop-down documents. The documents declaring HTML subwindows can never be drop-down. Only the leaf documents of a tree of subwindows may be drop-down, and are therefore
30 potential users of the method of the invention. Thus, the sum of the visible parts of the HTML documents that is able to be drop-down exactly covers the area reserved for the display of the HTML engine. This property is used to guarantee a use of the method of
35 the invention for each leaf document.

The HTML engine divides up the graphics memory

- 11 -

proportionately, as a function of the area of each subwindow:

- **M** represents the quantity of graphics memory reserved for the HTML engine,
- 5 - **Wm & Hm**: the size in pixels of the display window of the HTML engine,
- **Wf & Hf**: the size in pixels of any subwindow.

The quantity of graphics memory usable by this
10 subwindow to create its anticipation zones is proportional to the area:

$$Mf = M * (Wf * Hf) / (Wm * Hm)$$

An exemplary subwindow tree, in which each box
15 constitutes an HTML document, is illustrated in figure 4. There is thus a root document, for example of 600 × 400 pixels, which gives rise to a document Frame 1, for example of 600 × 100 pixels, and a document Frame 2, for example of 600 × 300 pixels, which gives
20 rise to a document Frame 2-1, for example of 200 × 300 pixels and a document Frame 2-2, for example of 400 × 300 pixels.

The display of such a tree gives the result illustrated
25 in figure 5.

In such an example the division of the graphics memory may be effected in the following manner, with, **M** = 1,920 Kbytes:

- 30 - Root document (declaration of frames) = 0 bytes
- "Frame 1" (leaf document) = 480 Kbytes
- "Frame 2" (declaration of frames) = 0 bytes
- "Frame 2.1" (leaf document) = 480 Kbytes
- "Frame 2.2" (leaf document) = 960 Kbytes

35

The following operation is a problem of determining the granularity of the pixmaps: for a quantity of graphics

- 12 -

memory available for a document it is necessary to determine what size and granularity of the pixmaps will allow the system to function properly. The objective is to guarantee at least one increment of scrolling, of anticipation of the motion on either side of the visible zone, and to have at least one row of pixmaps that can be displaced on one side or the other.

Figure 6 represents a complete HTML document 12 after page setup, the part covered by the pixmaps 13 and the visible part 11. This document 12 may at one and the same time have a horizontal scroll and a vertical scroll.

Represented in this figure are the following parameters:

- **Wd & Hd**: width and height in pixels of the complete document after page setup,
- **Wf & Hf**: width and height in pixels of the visible part of the document (size of the subwindow),
- **Wp & Hp**: width and height in pixels of a pixmap.

There are other parameters that are not represented in this figure:

- **Nx & Ny**: number of pixmaps horizontally or vertically,
- **Px & Py**: anticipation band guaranteed available horizontally and vertically on either side of the visible zone (this anticipation band corresponds, as a minimum, to the scroll increment),
- **T**: size of a pixel in bytes (dependent on the color coding system),
- **Mf**: quantity of graphics memory available for the pixmaps associated with the document.

It is firstly considered that the document 12 can be scrolled horizontally and vertically and that the

quantity of graphics memory is insufficient to cover the whole of the document with pixmaps both horizontally and vertically.

- 5 To be able to guarantee the permutation of the pixmaps and also at the same time the existence of a guaranteed anticipation band on each side, the following inequalities must be complied with:

10 $- 2 Px + Wf \leq Wp (Nx-1)+1$ (1)
with $Nx > 1$

$- 2 Py + Hf \leq Hp (Ny-1)+1$ (2)
with $Ny > 1$

15

The memory limitation also imposes a constraint. The sum of the pixmaps must not exceed the quantity of memory reserved:

20 $Mf \geq Nx*Ny*Wp*Hp*T$ (3)

The number of equations is still insufficient to determine the number and the size of the pixmaps. Hence, we opt for certain particular conditions:

- 25 - Px , the guaranteed horizontal anticipation zone is defined as equal to the horizontal scroll increment.
- Py , the guaranteed vertical anticipation zone is defined as equal to the vertical scroll increment.
- $Px = \alpha Wf$: the horizontal increment of scrolling is
30 proportional to the width of display of the document.
- $Py = \alpha Hf$: the vertical increment of scrolling is proportional to the height of display of the document with $0 < \alpha \leq 1$, α being a constant of the HTML engine (constraint guaranteeing that the whole of the
35 document can be consulted by incremental scrolling).
- $(Wp*Nx)*Hf = (Hp*Ny)*Wf$ (4): the dimensions of the anticipation zone are proportional to the dimensions

- 14 -

of the display zone.

The size and the number of pixmaps horizontally are determined using equations (3) and (4) and by considering that the maximum capacity of the graphics memory must be used. The following result is obtained:

$$W_p * N_x = ((Mf * Wf) / (Hf * T))^{1/2} \quad (5)$$

where $W_p * N_x$ corresponds to the width in pixels of the anticipation zone.

By using equations (1) and (5) we obtain the maximum width of a pixmap:

15

$$W_{pmax} = ((Mf * Wf) / (Hf * T))^{1/2} + 1 - (2 \alpha + 1) Wf$$

By dividing the width of the anticipation zone ($W_p * N_x$) of (5) by the maximum width of a pixmap W_{pmax} and by rounding up to the next integer, we obtain the minimum number of pixmap zones widthwise N_{xmin} .

For all of the values of $N_x \geq N_{xmin}$, the proper functioning of the method of the invention is assured. In the implementation of the HTML engine, it is the value N_{xmin} which is retained for the chopping. Specifically, at the decoder level, the number of pixmaps that can be created is limited. The possibility of using the method of the invention is thus guaranteed to the maximum and the attaining of a critical value such as this is therefore avoided.

The actual width of a pixmap W_p is obtained by dividing the width of the anticipation zone ($W_p * N_x$) of (5) by the retained number of pixmaps N_{xmin} rounded down to the next integer.

- 15 -

In the example of "Frame 2.2", by taking a coefficient α at 10% and $T = 4$ bytes, the results obtained are as follows:

- 5 - $N_x = 7$
- $W_p = 80$ pixels

The determination of the size and of the number of pixmaps vertically is carried out in the same manner as
10 horizontally.

In the example considered, the results obtained for "Frame 2.2" are as follows:

- 15 - $N_y = 7$
- $H_p = 60$ pixels

With this calculation scheme, the granularity of the pixmaps is always the same horizontally and vertically.

20

In the implementation of the decoder, it is necessary to take account of an additional parameter: the pixmaps created must always have dimensions that are multiples of 8 pixels. Consequently, after having determined the
25 size of each pixmap and the number of pixmaps in a direction, while rounding this size down to a multiple of 8, we check that equation (1) or (2) in the first direction processed is still satisfied. If this is no longer the case, the granularity of the pixmaps must be
30 increased by 1 and the new size of a pixmap must be recalculated.

With these roundings, part of the memory reserved for the calculation in a direction may no longer be fully
35 used. This memory surplus is then carried over to the other direction before calculating the size and the number of pixmaps. Under these new conditions, the

- 16 -

horizontal and vertical granularity may become different.

As illustrated in figure 7, in the case where
5 $W_p * N_x \geq W_d$, it is possible to cover the entire width of
the document with pixmaps. There is therefore no need
for a horizontal permutation mechanism. The
determination of the number and the granularity of the
10 pixmaps is different. The surplus of graphics memory
available at the horizontal scroll level is assigned to
the vertical scroll. The width of the pixmaps is then
the maximum width of creation of a pixmap in a decoder.